Aprendizado Não Supervisionado

Uarlley do Nascimento Amorim

Universidade Federal do Espirito Santo

28 de Fevereiro de 2022

Sumário

Supervisionado vs Não Supervisionado

Supervisionado

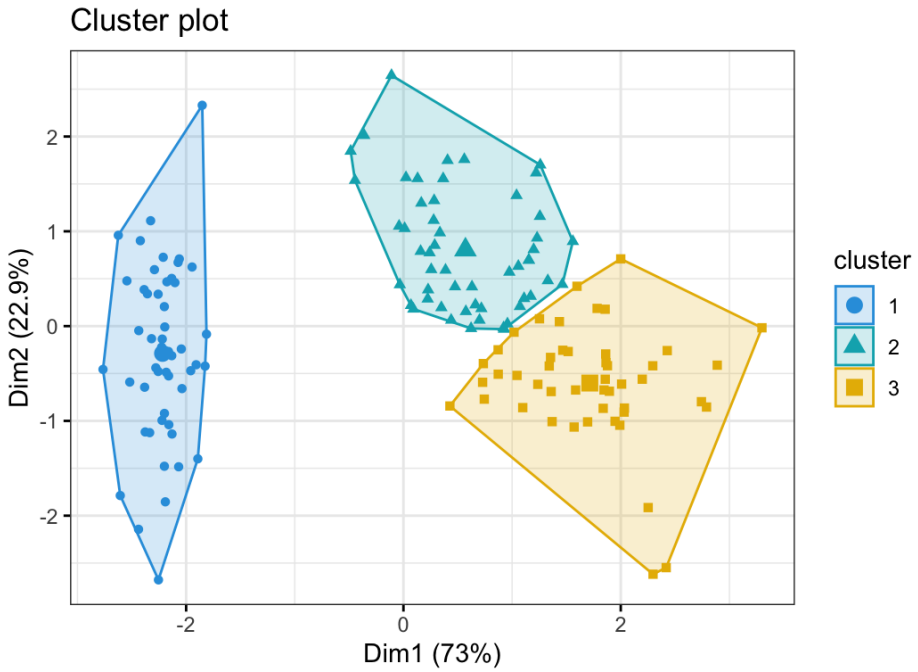
- Para qualquer conjunto de dados **X**, temos suas respectivas labels **Y**.
- **Objetivo:** Aprender uma *função* que mapeia $X \rightarrow Y$.
- **Exemplos:** Classificação, regressão, detecção de objetos, etc.

Não Supervisionado

- Temos apenas o conjunto de dados **X**, sem labels.
- **Objetivo:** Identificar padrões e similaridades na estrutura do conjunto de dados.
- **Exemplos:** Clusterização, redução dimensional, density estimation, modelos generativos, etc.

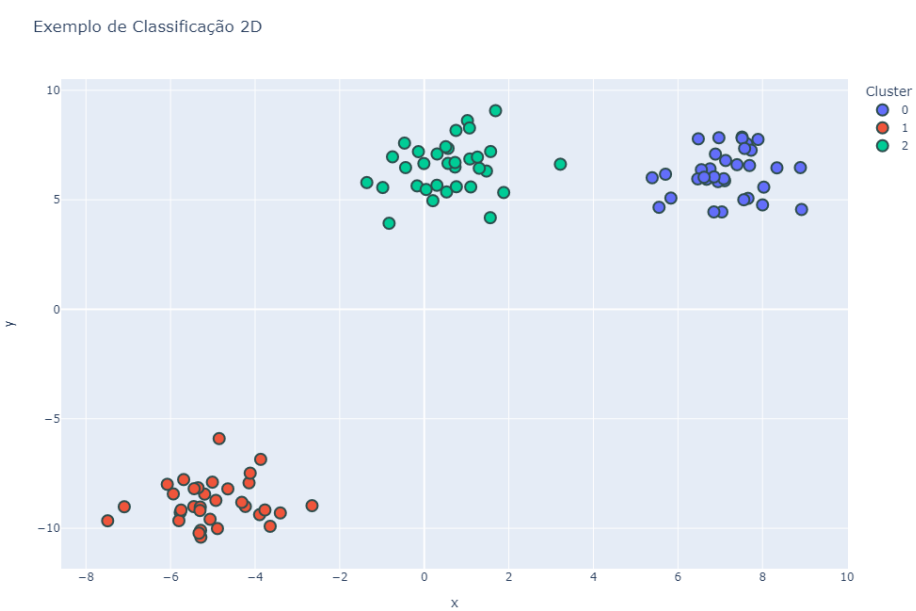
Clusterização

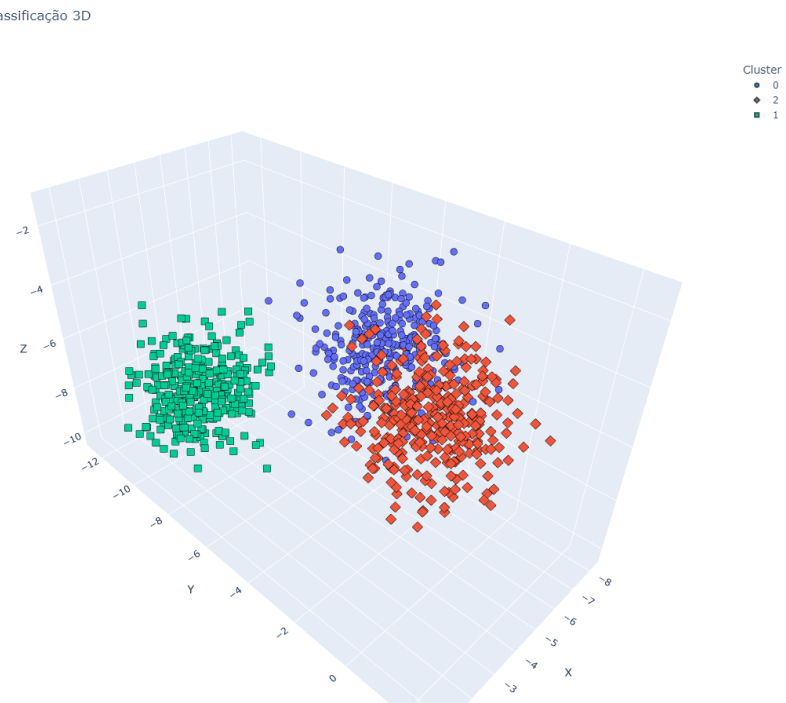
Tem como objetivo dividir um determinado conjunto de dados em ***n*** grupos a partir de uma determinada *similaridade*.



Clusterização

Geralmente, é possível agrupar um conjunto de dados a olho nu utilizando métodos de visualização.
Mas isso só é possível quando se trabalha com um conjunto de dados com $n \leq 3$ dimensões.





Clusterização

Nesse contexto, utilizamos modelos para automatizar este processo para conjuntos de dados ***n-dimensionais***.
Tais modelos identificam padrões nos dados, e, a partir disso, associam um dado á um determinado grupo.
O método de identificação de padrões pode variar de modelo à modelo, tais variações podem se dar em:

- Custo computacional;
- tempo de execução;
- performance;
- implementação.

Tipos de modelos

Baseados em distância

- Utiliza calculo de distância para associar um ponto ao centroide mais próximo.
- **Exemplos:** K-Means, Mean-shift e OPTICS.

Baseados em densidade

- Transforma regiões muito densas em clusters.
- **Exemplo:** DBSCAN.

Baseados em Hierarquia

- Agrupa os dados utilizando uma árvore de hierarquia.
- **Exemplo:** Hierarchical clustering.

Baseados em distribuição

- Assume que o conjunto de dados é um composto de distribuições.
- **Exemplo:** Gaussian Mixture Model.

Identificação de Outliers

Outliers são observações que estão bem distantes de outros dados em uma amostra qualquer de uma população.
Os outliers podem ser classificados em:

- Univariado: pontos extremos na distribuição de uma variável especifica. Ex: pessoa com 2.3 metros de altura.
- Multivariado: pontos improváveis em observação qualquer. Ex: Pessoas com 2 metros de altura.

Identificação de Outliers

As principais causas de outliers são:

- Erros na entrada dos dados (erro humano);
- erros de medida (erro instrumental);
- erros experimentais (erro de extração, planejamento ou execução);
- erros de processamento de dados (erro de manipulação);
- naturais (não é um erro e sim um dado improvável).

Para conjuntos de dados paramétricos, podemos utilizar métodos estatísticos para remover outliers, como por exemplo o método **IQR** (**intervalo interquartil**).

Já para dados não paramétricos, podemos utilizar modelos como **DBScans** e **Isolation Forest**.

Redução Dimensional

Quando se trabalha com conjuntos de dados com múltiplas dimensões, é importante utilizar métodos de redução tanto para visualização, quanto para aplicação de modelos.

Os métodos são:

- PCA: utiliza projeções lineares, ou seja, quanto mais linear for a relação entre as variáveis, melhor será o desempenho do PCA, ou melhor, menos informação será perdida após a redução.
- TSNE: utiliza projeções não linerares. Busca reduzir as dimensões tentando preservar a vizinhança dos pontos.

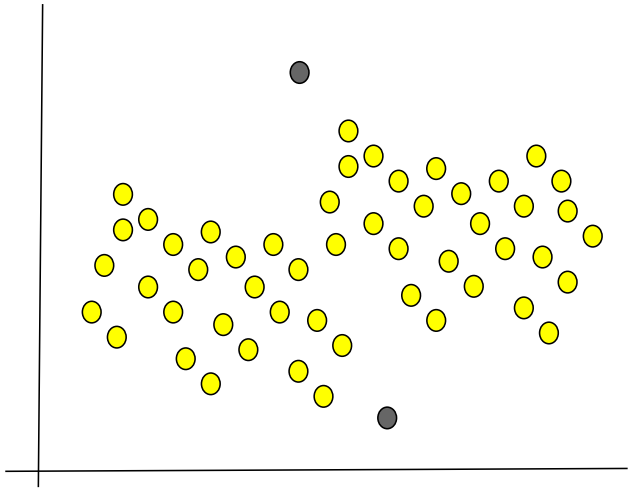
TSNE vs PCA

PCA	TSNE
Busca preservar a estrutura global dos dados	Busca preservar a estrutura local dos dados
Não depende de hyperparametros	Depende de Hyperparametros como perplexity e early exageration.
É sensivel a outliers	Consegue lidar bem com outliers
Funciona a partir da rotação de vetores preservando a variância.	Busca minimizar a distância entre os pontos em uma gaussiana.
É possivel determinar o quanto de variância queremos manter	Não podemos preservar a variância dos dados.
Possui baixo custo computacional	Possui um alto custo computacional.

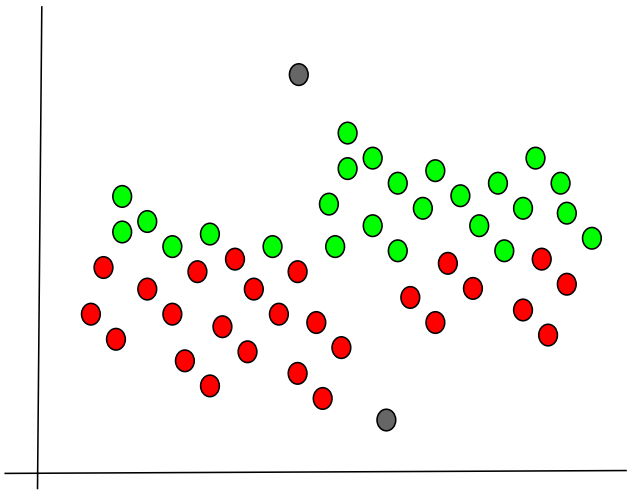
K-Means

- Tem como objetivo encontrar similaridades entre os dados e agrupá-los em um número de grupos definido a priori.
- Utiliza um método simples e eficiente baseado no conceito de distância, atribuindo de forma iterativa os pontos de dados ao grupo que representa a menor distância.
- Algumas aplicações são:
  - Identificar perfil de clientes
  - Segmentação de mercado
  - Visão computacional
  - Motores de busca
  - Astronomia

Algoritmo

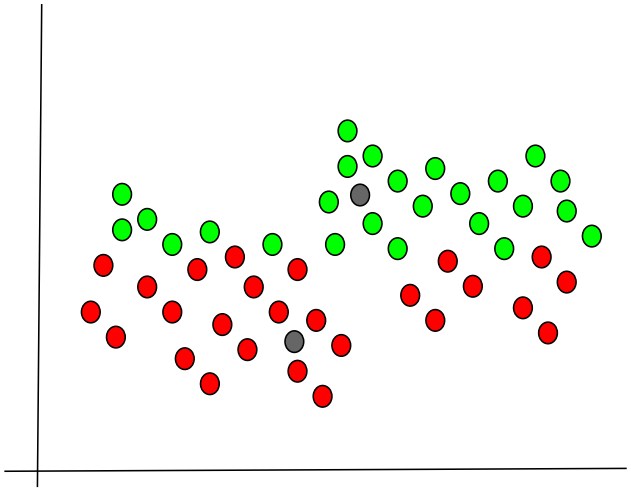


Cria dois pontos aleatórios do espaço, chamados centróides.

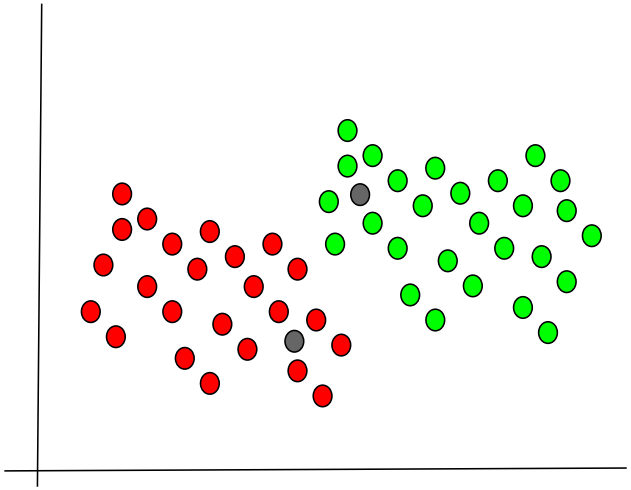


Associa cada ponto ao centróide mais próximo.

Algoritmo



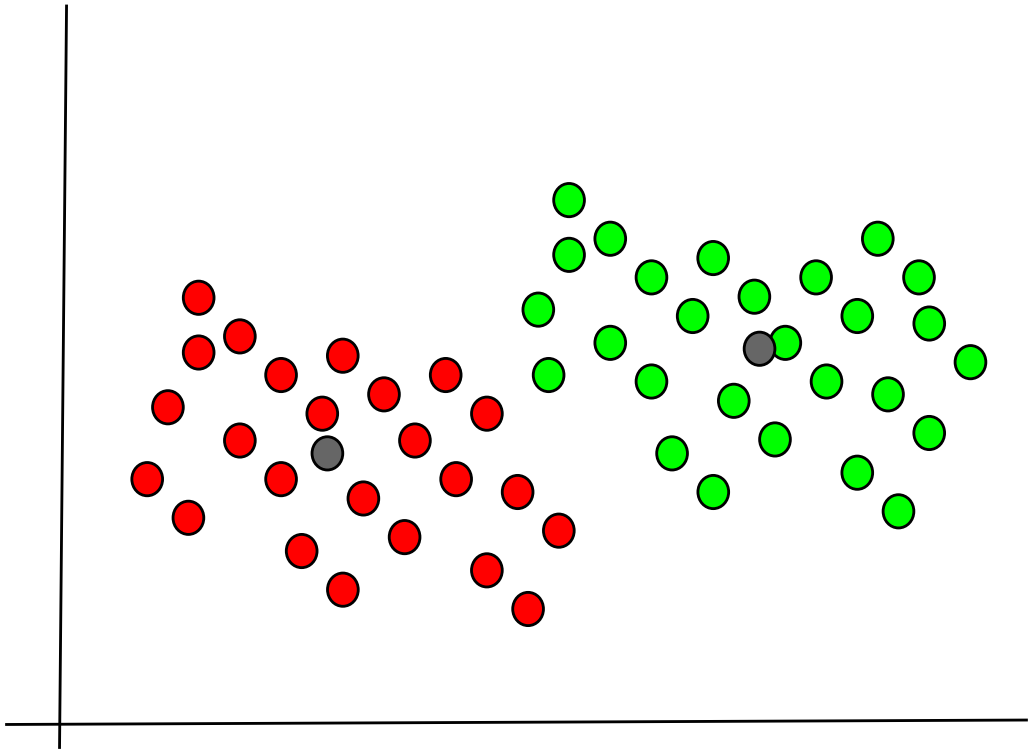
Move os centróides para a posição média dos pontos pertencentes à ele.



A distância é recalculada e os pontos são novamente associados ao centróide mais próximo.

Algoritmo

O processo é repetido iterativamente, até que o reposicionamento do centróide seja insignificante.



Exemplo

Exemplo de Utilização em Python:

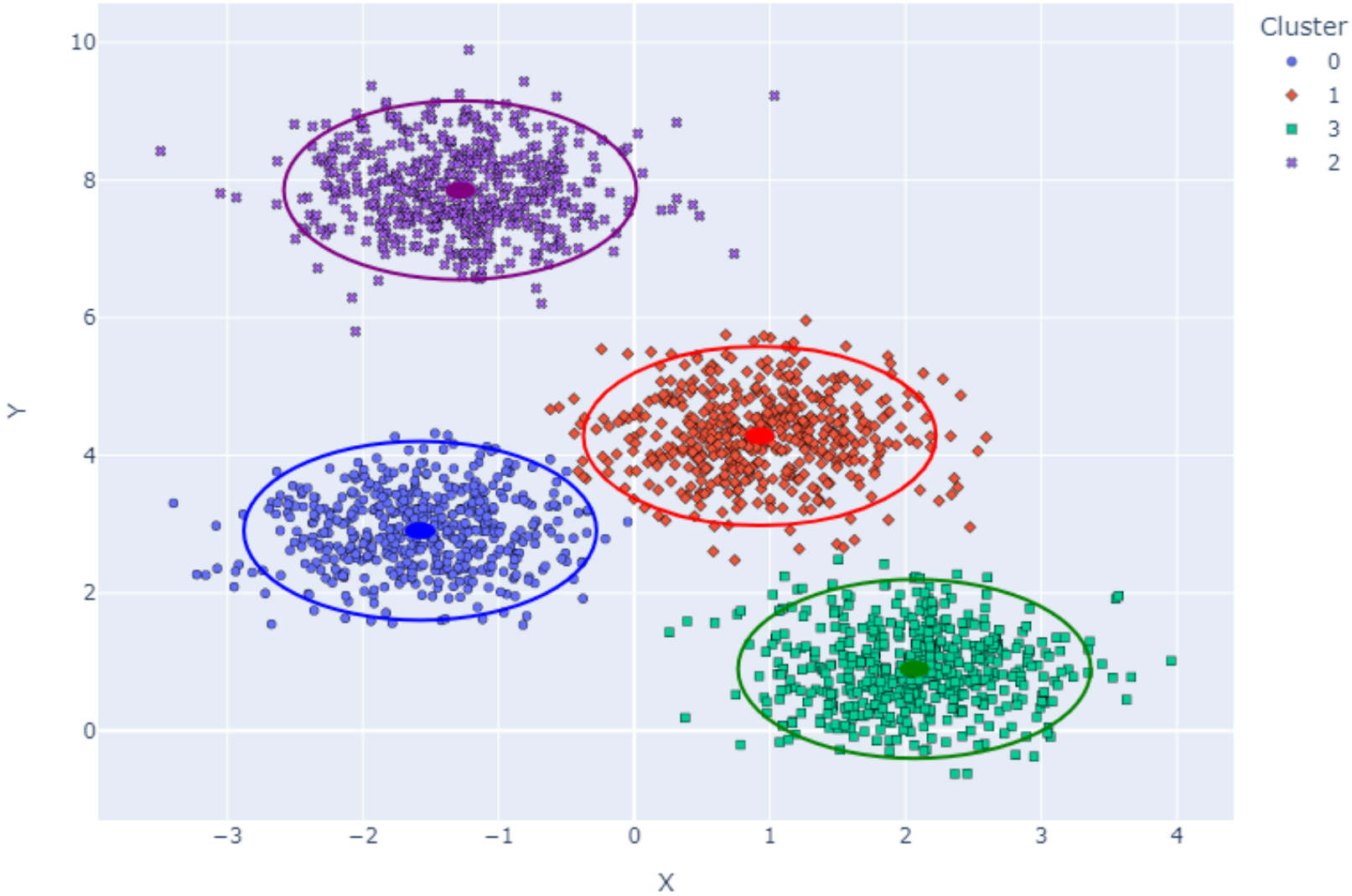
```
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=2000, centers=4, cluster_std
    =0.60, random_state=0)

data = pd.DataFrame({"X": X[:,0], "Y": X[:,1]})

km = KMeans(n_clusters=4)
labels = km.fit(data).predict(data)
```

Exemplo



Como encontrar o melhor K

As vezes estaremos trabalhando com um conjunto de dados com múltiplas dimensões, e, nestes casos, se torna impossível visualizar e identificar a quantidade de clusters ideal. Nesse contexto, podemos utilizar alguns métodos para identificar qual é a melhor quantidade de clusters.

Os principais são:

- Método do Cotovelo
- Coeficiente de Silhouette

Método do Cotovelo

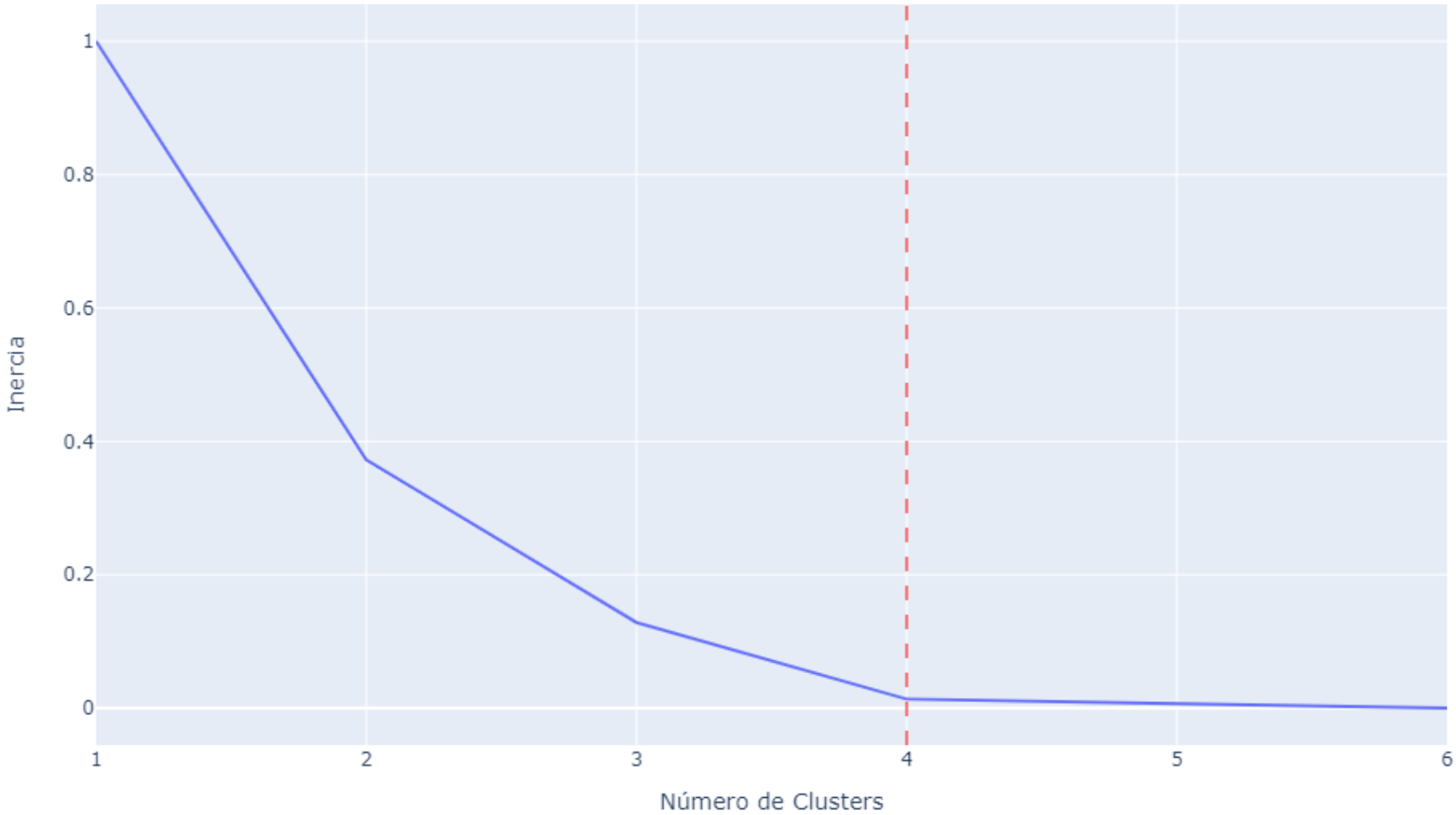
O Método do Cotovelo ou A Curva de Cotovelo é uma técnica usada para encontrar a quantidade ideal de clusters K. Para isso, o método:

- Testa a variância dos dados em relação ao número de clusters e a inércia do modelo.
- Define como cotovelo o momento em que variância tende a ser menor, ou seja, quando a não existe uma discrepância significativa.

Como a partir deste ponto a variância é baixa e queremos escolher a menor quantidade de clusters possível, tal quantidade seria exatamente onde o cotovelo estaria.

Método do Cotovelo

Método aplicado no exemplo anterior:



Coeficiente de Silhouette

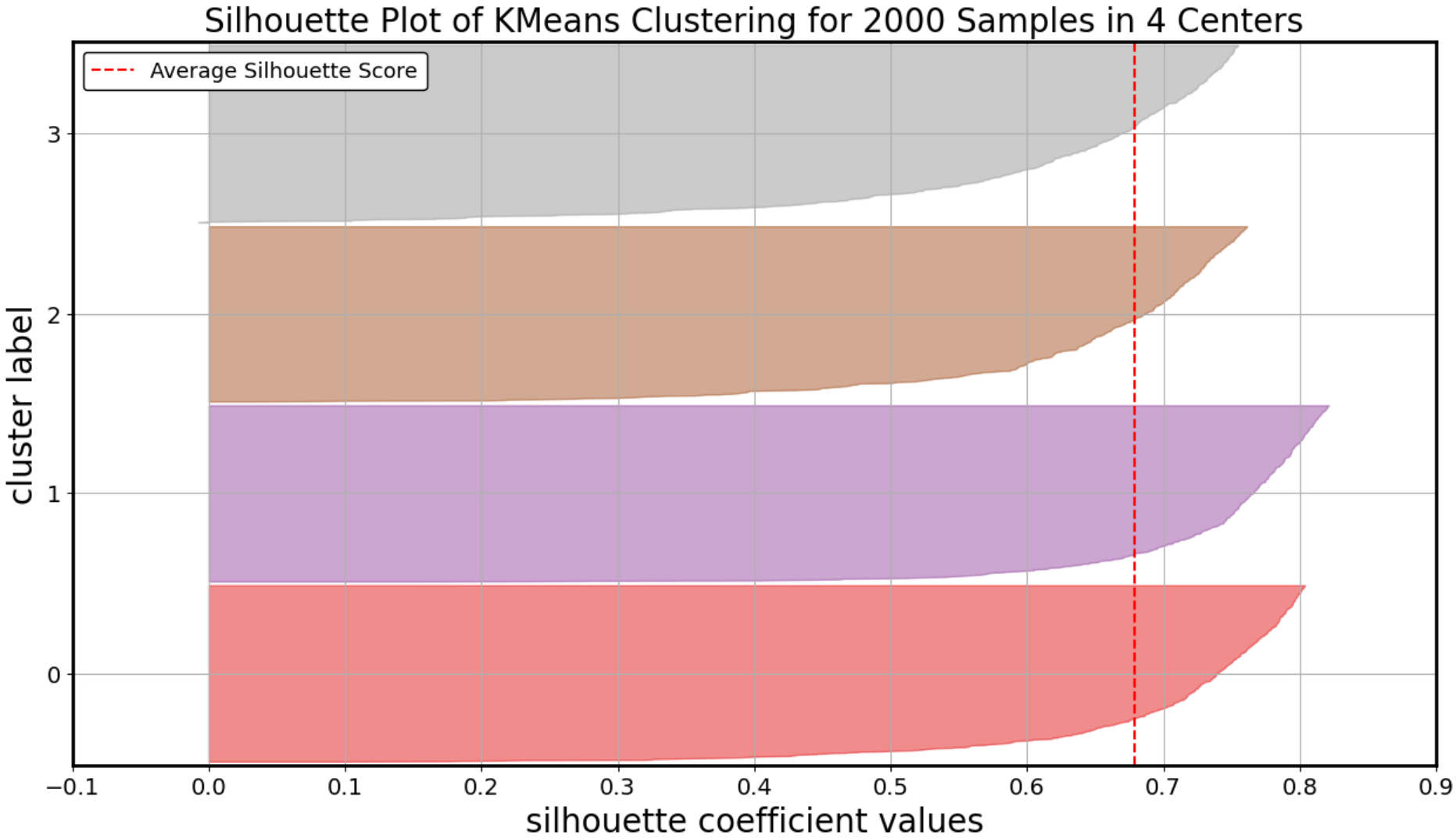
O Coeficiente de Silhouette é calculado a partir da distância média entre os pontos de um cluster (a), e a distância média entre os clusters (b). A partir disso, o coeficiente é calculado utilizando seguinte fórmula:

$$\frac{(b-a)}{\max(a,b)} \tag{1}$$

O coeficiente varia de 1 à -1. Se o coeficiente é 1, os clusters estão densos e bem separados. Um valor próximo de 0 significa que os clusters estão sobrepostos. Um coeficiente negativo representa que existem pontos associados com o cluster errado.

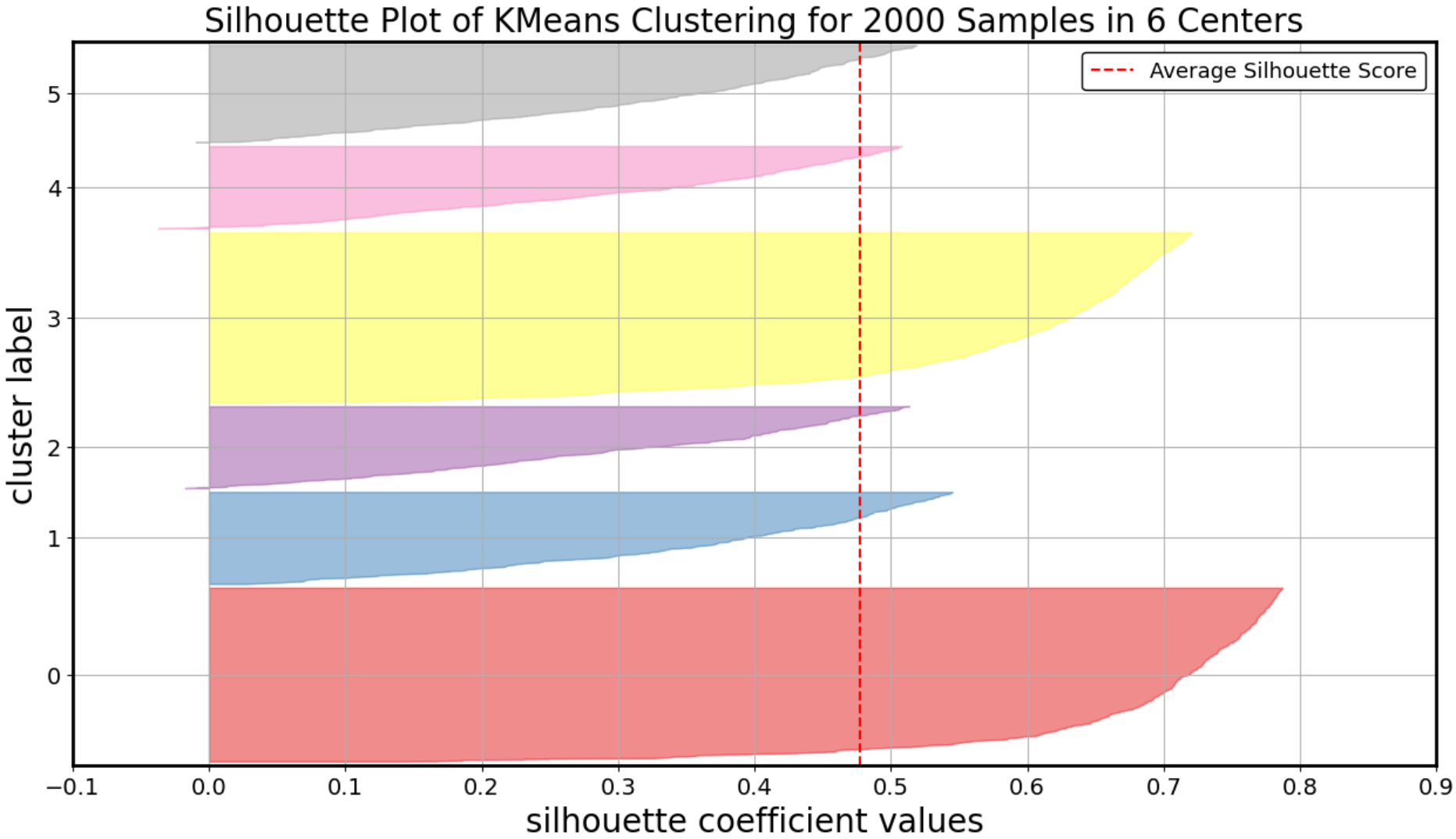
Coeficiente de Silhouette

Coeficiente de Silhouette aplicado no exemplo anterior com $K = 4$:



Coeficiente de Silhouette

Coeficiente de Silhouette aplicado no exemplo anterior com $K = 6$:



Identificando Outliers a partir do K-Means

Como mencionado anteriormente, pode acontecer de termos pontos que supostamente deveriam pertencer à um cluster a, porém, o modelo o classificou como b. Este tipo de problema é muito comum em problemas de aprendizado não supervisionado.

Uma boa estrategia, é simplesmente não classificar este tipo de dado, evitando falsos positivos/negativos.

Tal abordagem é simples de ser realizada no K-Means, já que o mesmo utiliza distância para classificar os dados.

Identificando Outliers a partir do K-Means

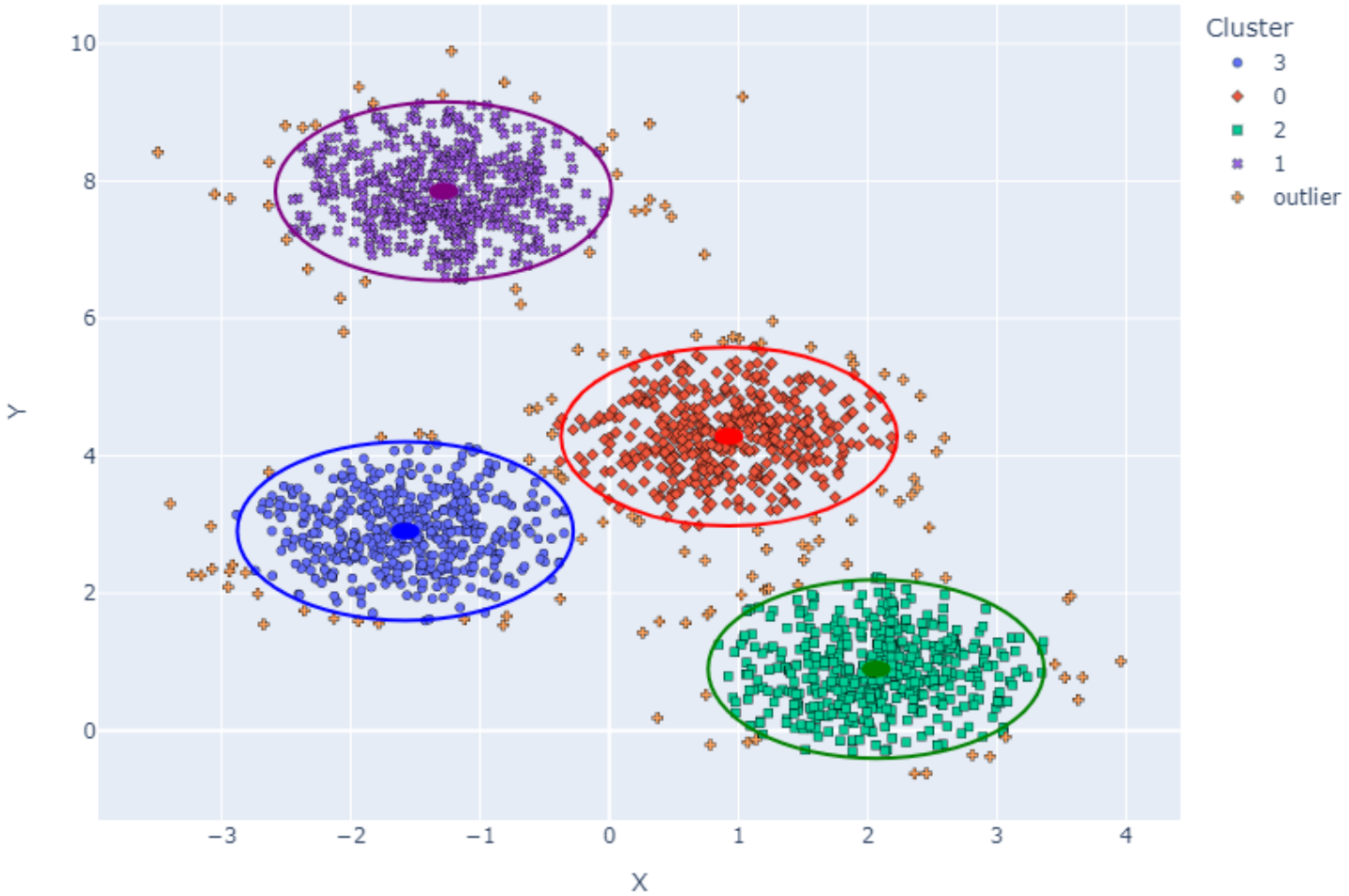
Tal processo pode ser feito da seguinte forma:

1. Coleta as distâncias de um ponto à todos os centroides.
2. Coleta a menor das distâncias e a normaliza.
3. Estabelece um limitante α entre 0 à 1.
4. Se a distância normalizada for maior que α, não classifique, caso contrario, classifique.

O resultado deste método é a identificação dos pontos que ”não sabem onde se encaixam”, ou seja, pontos ”indecisos”.

Identificando Outliers a partir do K-Means

Aplicação do método no exemplo anterior, com $\alpha = 0.5$.

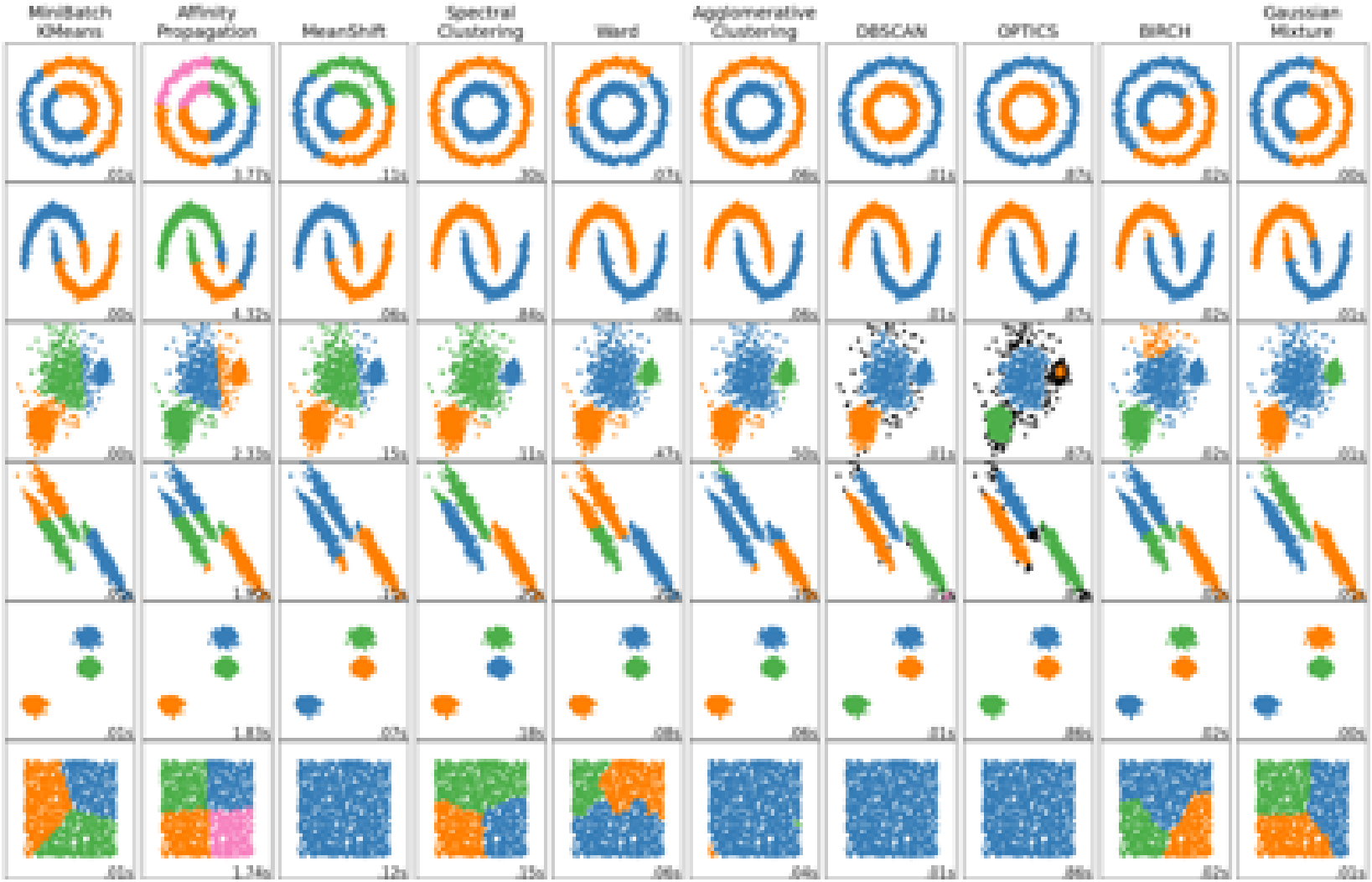


Fraquezas

- É necessário determinar o numero de clusters K previamente.
- Tem problemas com conjuntos de dados com diferentes escalas.
- Não lida bem com clusters não convexos.

A imagem a seguir compara o desempenho de modelos de aprendizado não supervisionado em conjuntos de dados com clusters convexos e não-convexos.

Fraquezas


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture